



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/529,638	11/04/2005	Josef Dietl	09282.0058-00	1384
22852	7590	02/25/2010	EXAMINER	
		FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP 901 NEW YORK AVENUE, NW WASHINGTON, DC 20001-4413	KRETZMER, ERIKA A	
			ART UNIT	PAPER NUMBER
			2192	
			MAIL DATE	DELIVERY MODE
			02/25/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/529,638	DIETL ET AL.	
	Examiner	Art Unit	
	Erika Kretzmer	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 25 November 2009.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-8 and 10-21 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-8 and 10-21 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 31 March 2005 and 14 November 2006 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____ .
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)	5) <input type="checkbox"/> Notice of Informal Patent Application
Paper No(s)/Mail Date _____. _____	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. The following is a Final Office action in response to applicant's amendment and response received November 25, 2009, responding to the August 28, 2009 office action provided in rejection of claims 1-8 and 10-21.
2. Claims 1, 2, 12, and 13 have been amended. Claims 1-8 and 10-21 are pending and are addressed in this office action.
3. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR §1.136(a).
4. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Response to Arguments

5. Applicant's arguments filed 11/25/2009, in particular pages 9-14, have been fully considered.
6. Applicant's amendments to the claims filed 11/25/2009 present substantial new limitations (at least claim 1 lines 5-7 and claim 1 lines 9-13) which raises the issue of new matter under 35 USC 112, first paragraph. Examiner is not readily able to find support for the new limitations in the specification as amended 11/25/2009 or as originally filed. Applicant is required to point out where in the specification each new limitation finds support, in order to avoid the issue of new matter under 35 USC 112, first paragraph.

7. With respect to the **objection to the specification**, the amended specification paragraphs received 11/25/2009 overcome this objection and it is withdrawn. Examiner finds support for the amendments to the specification, at least in the claims as originally filed.
8. Examiner interprets further references to an “actual” implementation in the specification, such as “the actual implementation of the provider class” (page 2 lines 15-16), “the implementation” (page 2 line 30) and “implementation class description” (page 3 lines 25-26) to be equivalent to the definition module and its class. This interpretation is consistent with the amendments to the specification as filed, and the claims as amended 11/25/2009.
9. With respect to the **double patenting rejection of the claims**, applicant argues that no actual double patenting situation has arisen. However, a Notice of Allowance was sent to Applicant on 6/4/2009, the issue fees were paid by Applicant on 7/28/2009, and an Issue Notification for Patent 7,584,457 was sent to Applicant on 8/12/2009. This patent was published as issued on 9/1/2009. Therefore, the provisional nature of this rejection is withdrawn, and the rejection is maintained.
10. With respect to the **rejections of the claims under 35 USC 112, second paragraph**, the amendment to the specification overcomes these rejections and they are withdrawn.
11. With respect to the **rejection of the claims under USC 103**, applicant argues that certain features of the newly amended claims are not shown by the combination of McLaughlin and Lucas. Applicant’s arguments have been fully considered and are not persuasive.
12. Applicant argues that neither McLaughlin nor Lucas show “the validation tool determining whether **the class is in compliance with the interface** and **whether the method of the interface can be used to execute the runtime function** when the object is called during runtime execution of the computer program”, emphasis added by Applicant. This limitation is found in claim 1 and 12 as amended 11/25/2009. However, McLaughlin shows this limitation. Applicant states on page 3, beginning line 6: “a program is programmed in object oriented style using two separate tree structures, written in XML, wherein the first tree structure represents the classes to be implemented and the second tree structure represents the associated interfaces.” Applicant further states on page 3, beginning line 24: “On the XML level a syntax check is performed

between the interface description and the implementation class description". McLaughlin teaches validating the set of definition instructions ("XML document") and implementation instructions ("constraint model") using a validation tool (see at least section 3.1.1, particularly "ensure that your constraint model syntax is supported by the binding framework you want to use" and "Write several XML documents ... and validate them against your new constraints."). Finally, Applicant teaches that the final validation is done through a compiler in at least page 4 lines 4-34, particularly: "The validity of these interfaces and classes is then proved using methods as described above, i.e. using a compiler that performs the usage and implementation checks at the semantics level." McLaughlin also teaches validating whether the class is in compliance with the interface and whether the method of the interface can be used to execute the runtime function, in at least figure 3-1, pages 13 and 14 "Class generation process flow", particularly "Java compiler." Applicant further admits on page 4 lines 21-22 of the specification as filed, "Such interpreters or parsers are known per se in the art." Therefore, the combination of McLaughlin and Lucas does show these limitations.

13. Applicant argues that neither McLaughlin nor Lucas show "validating... [by] determining whether the class is in compliance with the interface and whether the method of the interface can be used to execute the runtime function..., wherein **the determination is made before runtime execution of the computer program and during compilation of the computer program,**" (emphasis added) as further recited in claims 1 and 12 as amended 11/25/2009. However, McLaughlin shows this limitation. Applicant teaches that the final validation is done through a compiler in at least page 4 lines 4-34, particularly: "The validity of these interfaces and classes is then proved using methods as described above, i.e. using a compiler that performs the usage and implementation checks at the semantics level." McLaughlin also teaches validating before runtime and during compilation, in at least figure 3-1, pages 13 and 14 "Class generation process flow", particularly "Java compiler." Applicant further admits on page 4 lines 21-22 of the specification as filed, "Such interpreters or parsers are known per se in the art." Therefore, the combination of McLaughlin and Lucas does show these limitations.

Art Unit: 2192

14. The combination of McLaughlin and Lucas shows these limitations. Therefore, the independent claims remain rejected under 35 USC 103 over McLaughlin and Lucas. The dependent claims are rejected at least because they depend from rejected claims.
15. Examiner notes that Applicant has provided no further argument with respect to the art applied to the claims.

Response to Amendment

16. Applicant's amendments to the claims filed 11/25/2009 present substantial new limitations which raises the issue of new matter under 25 USC 112, first paragraph. Examiner is not readily able to find support for the new limitations in the specification as amended 11/25/2009 or as originally filed. Applicant is requested to point out where in the specification these new limitations find support, in order to avoid the issue of new matter under 25 USC 112, first paragraph.
17. Applicant's cancellation of the claims and amendments to the specification overcome the objection and rejections set forth in the previous Action. However, applicant's newly presented claims do not overcome the entire art of record. New rejections are presented herein.

Status of Claims

18. This action is in reply to the amended claims filed on 11/25/2009. Claims 1-8 and 10-21 are currently pending and have been examined.
19. This application claims priority to PCT/EP03/10442 filed on September 18, 2003. This application claims priority to EP 0202204.2 filed on October 1, 2002. A certified copy of the foreign application was received by the Office.

Specification

20. The amendment to the specification filed 11/25/2009 has been entered because it finds support at least in the original claims. Examiner interprets further references to an "actual" implementation in the specification to be equivalent to the definition module and its class (such as "the actual implementation of the provider class" (page 2 lines 15-16), "the implementation" (page 2 line 30) and "implementation class description" (page 3 lines 25-26)). This interpretation is consistent with the amendments to the specification as filed.

21. The original specification filed 3/31/2005 was published as pre-grant publication US 2006/0248538 A1. The original specification was amended by Applicant at the time of filing on 3/31/2005, and further amended on 1/25/2009. The specification is accepted as amended.

Drawings

22. Original drawing 1 was received on March 31, 2005. Replacement drawing 1 was received on November 14, 2006. Examiner notes that the replacement drawing was described in the amendment to the specification received 3/31/2005.

23. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Seven features that admit of illustration were newly added to the claims by amendment on 11/25/2009. The following seven features must be shown in the drawings or the features canceled from the claims:

- the set of definition instructions includes a class having an object with a runtime function,
- the set of implementation instructions includes an interface having a method,
- the validation tool determining whether the class is in compliance with the interface,

Art Unit: 2192

- the validation tool determining whether the method of the interface can be used to execute the runtime function when the object is called during runtime execution of the computer program,
- runtime execution of the computer program,
- compilation of the computer program, and
- wherein the determination is made before runtime execution of the computer program and during compilation of the computer program.

24. Examiner suggests that the first two features may be shown in a new drawing, and the latter five features may be added to the flow chart of Figure 1. No new matter should be entered. Applicant is required to point out where in the specification each newly illustrated feature of the claims finds support, in order to avoid the issue of new matter under 35 USC 112, first paragraph.

25. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Double Patenting

26. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).
27. A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.
28. Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).
29. **Claims 1, 2, 12, and 13** are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1 and 4 of U.S. Patent No. 7,584,457. Although the conflicting claims are not identical, they are not patentably distinct from each other because the pending claims are obvious in view of the claims of the patent. For example, the following table compares claim 1 of the present application to claim 1 of the commonly owned patent:

Present application (10/529638)	US Patent No. 7,584,457
<p><i>1. A computer-implemented method for validating computer code the method being executed by a computer and, comprising:</i></p> <p><i>providing a computer program by defining</i></p> <p><i>at least one set of implementation instructions ... wherein the set of implementation instructions includes an interface having a method</i></p> <p><i>at least one set of definition instructions, wherein the set of definition instructions includes a class having an object with a runtime function</i></p> <p><i>validating the set of definition instructions and the set of implementation instructions using a validation tool</i></p> <p><i>the validation tool determining whether the class is in compliance with the interface and whether the method of the interface can be used to execute the runtime function when the object is called during runtime execution of the computer program,</i></p> <p><i>wherein the determination is made before runtime execution of the computer program and during compilation of the computer program</i></p> <p><i>and at least a script code section;</i></p> <p><i>validating the script code section using the set of implementation instructions.</i></p>	<p>1. A method for validating programs, the method comprising steps implemented by one or more computers of: (lines 1-2)</p> <p>receiving a meta-language description of a computer program, the meta-language description comprising (lines 3-4)</p> <p>a meta-language definition module (lines 4-5) ... the meta-language definition module defining a first interface associated with the class; (lines 8-10)</p> <p>and a meta-language implementation module, the meta-language implementation module defining a first class to be implemented by the computer program (lines 5-8)</p> <p>validating the meta-language description by validating syntax of the meta-language definition module and syntax of the meta-language implementation module; (lines 11-13)</p> <p>generating a language-dependent program from the meta-language description, the language-dependent program comprising the first interface, the first class (lines 14-16)</p> <p>performing usage and semantic checks on the computer program by compiling the generated first interface and the generated first class; (lines 18-20)</p> <p>and a script code section written in a scripting language (lines 16-17)</p> <p>performing usage checks on the script code section by extracting language elements from the generated script code section and comparing the extracted language elements with the meta-language definition module used to generate the language-dependent program. (lines 21-25)</p>

30. Note that "providing a computer program", as in claim 1 of the instant application, may be interpreted as "receiving a meta-language description of a computer program" (see at least specification page 3 lines 6-26). See also the subsequent claim limitations (e.g. claim 4), which define meta-language (XML) instructions which comprise the provided computer program. Compare to the "XML" "tree structures" used to define the meta-language module in US Patent No. 7,584,457, column 3, lines 14 through 45.

31. Note that in the present application, the "definition instructions" include "classes" in claim 1, and the "implementation instructions" include "interfaces" in claim 1. Claim 1 of US Patent No. 7,584,457 include "a first class" in an "implementation module", and "a first interface" in a "definition module." Therefore, the "definition instructions" of the present application have an equivalent function to the "implementation module" of US Patent No. 7,584,457. This terminology is further supported by amendments to the specification filed 11/25/2009.

32. Thus, "validating the script code with the implementation instructions," as in the present claim 1, is equivalent to "comparing the extracted language elements with the meta-language definition module", as in claim 1 of US Patent No. 7,584,457.

33. For these reasons, the limitations of claim 1 would have been obvious to one of ordinary skill in the art at the time the invention was made in view of claim 1 of US Patent No. 7,584,457.

34. Claim 2 adds the limitations of definition modules having "a plurality of classes" and implementation modules having "a plurality of interfaces," emphasis added. Claim 1 of the conflicting patent recited with "the first class" and "the first interface". Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made that the instructions could include a plurality of classes and a plurality of interfaces because it would allow the software to accomplish more than one function.

35. Claims 12 and 13 are computer readable medium versions, which recite the same limitations of claims 1 and 2. A functionally similar computer program product is found claim 4 of US Patent No. 7,584,457.

36. **Claims 3-6, 10-11, 14-17, and 20-21**, are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1 and 4 of US Patent No. 7,584,457 in view of “Java and XML Data Binding” (McLaughlin, 2002). Although the conflicting claims are not identical, they are not patentably distinct from each other because they would have been obvious over the copending claims in view of the cited art.

37. As to claims 3 and 14, in addition to the limitations of the parent claims being wholly anticipated by the claims of US Patent No. 7,584,457 as discussed above, McLaughlin further teaches the *definition instructions are converted into classes* (see at least section 3.1.4, figure 3-1) *and the set of implementation instructions are converted into interfaces* (see at least section 6.4.4).

38. As to claims 4, 6, 10-11, 15, 17, and 20-21, in addition to the limitations of the parent claims being obvious over the claims of US Patent No. 7,584,457 as discussed above, McLaughlin further teaches *the set of definition instructions (“XML documents”) and the set of implementation instructions (“XML constraints”) are described in XML* (see at least section 2.3.1, particularly “XML file” and figure 2-2 “XML documents” and section 2.3, particularly “XML constraints”). It is readily apparent that McLaughlin teaches the files are in a tree structure because the files are XML files. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the meta-language instructions of claim 1 of US Patent No. 7,584,457 with the XML of McLaughlin because XML (“eXtensible Markup Language”) is a common type of meta-language.

39. As to claims 5 and 16, in addition to the limitations of the parent claims being obvious over the claims of US Patent No. 7,584,457 and cited art as discussed above, McLaughlin further teaches *the classes and the interfaces are defined in Java language* (see at least section 3.1.4, figure 3-1, particularly: "The result of the generation step is one or more Java source files"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the language-dependent program of the claims 1 and 11 of US Patent No. 7,584,457 with the Java language of McLaughlin because Java is a common language for language-dependent programs.

40. **Claims 7-8 and 18-19**, are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1 and 4 of the claims of US Patent No. 7,584,457 in view of Lucas et al. (US 6,754,884 B1). Although the conflicting claims are not identical, they are not patentably distinct from each other because they would have been obvious over the copending claims in view of the cited art.

41. As to claims 7 and 18, in addition to the limitations of the parent claims being obvious over the claims of US Patent No. 7,584,457 as discussed above, Lucas further teaches that *the script code section is JavaScript* (see at least column 3, lines 45-55, particularly "a scripting language, such as JavaScript"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the scripting language of claim 1 of US Patent No. 7,584,457 with the JavaScript of Lucas because JavaScript is a common scripting language.

Art Unit: 2192

42. As to claims 8 and 19, in addition to the limitations of the parent claims being obvious over the claims of US Patent No. 7,584,457 and cited art as discussed above, Lucas further teaches that *validating the script code section comprises generating a symbol table by executing the code section in an interpreter* ("parser"), and *comparing the symbol table with the implementation instruction* ("XML data type declarations") (see at least column 3, line 56 through column 4 line 7, particularly: "a JavaScript-aware parser (e.g. parser 105) is equipped to recognize XML data type declarations and associate them with the appropriate items in the corresponding symbol table"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the validation of claim 1 of US Patent No. 7,584,457 with the validation using the symbol table of Lucas because it would allow validating a script-based web service with XML constraints (column 3 lines 45-55).

Claim Rejections - 35 USC § 112

43. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

44. Claims 1 and 12 are accepted under 35 USC 112, second paragraph in view of the amended specification paragraphs received 11/25/2009. Examiner interprets further references to an "actual" implementation in the specification, such as "the actual implementation of the provider class" (page 2 lines 15-16), "the implementation" (page 2 line 30) and "implementation class description" (page 3 lines 25-26) to be equivalent to the definition module and its class. This interpretation is consistent with the amendments to the specification as filed.

45. Claims 5 and 16 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Art Unit: 2192

46. Claims 5 and 16 are rejected under 35 USC 112, second paragraph in view of the amendments to claims 1 and 12 received 11/25/2009. Claim 5 depends from claims 4, 3 and 1. A new limitation of claim 1 recites "the set of definition instructions includes a class having an object with a runtime function." Examiner notes that definition instructions are described in page 3 line 6-26 as an XML file. Examiner interprets the reference to classes in claim 1, as classes defined in XML as described in page 3 line 6-26. Claim 4 further limits the definition instructions to being described in XML. Claim 3 adds the limitation of "the set of definition instructions are converted into classes." Examiner interprets this limitation as occurring as described on page 4 lines 5-10. Therefore, the XML instructions including a class are further converted into new classes. However, claim 5 recites, "wherein the classes ... are defined in the Java language." This limitation is unclear because it is not clear whether this limitation refers to the classes recited in claim 1 or the classes recited in claim 3.

47. A second, analogous, problem is found in claim 5 with respect to the limitation of claim 5 "interfaces are defined in the Java language." Compare that limitation with the "set of implementation instructions includes an interface" (claim 1), "the set of implementation instructions are converted into interfaces" (claim 3), "set of implementation instructions are described in XML (claim 4) and "interfaces are defined in the Java language" (claim 5). This limitation is unclear because it is not clear whether this limitation refers to the interfaces recited in claim 1 or the interfaces recited in claim 3.

48. Under the principles of compact prosecution, examiner treats the "classes" and "interfaces" in claim 5 as referring to the "classes" and "interfaces" in claim 3.

49. Claim 16 recites similar limitations to claim 5 and is rejected for the same reason.

Claim Rejections - 35 USC § 103

50. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

51. Claims 1-8 and 10-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Java and XML Data Binding" (McLaughlin, 2002) in view of Lucas et al. (US 6,754,884 B1).

Claim 1

McLaughlin teaches a *computer-implemented method for validating computer code* (see at least section 3.1.4, Figure 3-1 validating Binding schema against DTD). McLaughlin further teaches *the method being executed by a computer* (see at least page 13, sidebar 1, "at runtime"). McLaughlin further teaches *providing a computer program by defining at least one set of definition instructions* ("binding schema") and *at least one set of implementation instructions* ("constraint model") (see at least section 3.1).

McLaughlin further teaches *the set of implementation instructions* ("constraint model") *includes an interface* ("element") *having a method* ("attribute", "ATTRLIST") (see at least section 3.2, "Creating the constraints", particularly "a set of constraints ready to generate classes from" and example 3-1 "ELEMENT movies" and "ATTRLIST movies version"). McLaughlin further teaches *the set of definition instructions* ("binding schema") *includes a class having an object with a runtime function* (see at least section 3.3, particularly: "Once you've got your constraints ... you're ready to create a binding schema for your classes. This will instruct the class generation tool to generate classes, to use a specific Java package, to use collections, and a variety of other options.", example 3-4 "Modified binding schema for movies database", particularly, "type='class'" and "The result of this addition is apparent in the Movies class, which has multiple Movie subobjects").

McLaughlin further teaches *validating the set of definition instructions* ("XML document") and *the set of implementation instructions* ("constraint model") using a validation tool (see at least section 3.1.1, particularly "ensure that your constraint model syntax is supported by the binding framework you want to use" and "Write several XML documents ... and validate them against your new constraints.")

McLaughlin further teaches *the validation tool determining whether the class is in compliance with the interface* (see at least section 3.4.3 "Verifying Output", particularly: "To ensure that the generated classes work, all you need to do is ensure that they compile"). Applicant teaches that the final validation is done through a compiler in at least page 4 lines 4-34, particularly: "The validity of these interfaces and classes is then proved using methods as described above, i.e. using a compiler that performs the usage and implementation checks at the semantics level." McLaughlin also teaches validating before runtime and during compilation, in at least figure 3-1, pages 13 and 14 "Class generation process flow", particularly "Java compiler."

McLaughlin further teaches *the validation tool determining whether the method of the interface can be used to execute the runtime function when the object is called during runtime execution of the computer program* (see at least section 3.4.3 "Verifying Output", particularly: "To ensure that the generated classes work, all you need to do is ensure that they compile"). Applicant teaches that the final validation is done through a compiler in at least page 4 lines 4-34, particularly: "The validity of these interfaces and classes is then proved using methods as described above, i.e. using a compiler that performs the usage and implementation checks at the semantics level." McLaughlin also teaches validating before runtime and during compilation, in at least figure 3-1, pages 13 and 14 "Class generation process flow", particularly "Java compiler."

McLaughlin further teaches *wherein the determination is made before runtime execution of the computer program and during compilation of the computer program* (see at least figure 3-1, pages 13 and 14 "Class generation process flow", particularly "Java compiler").

McLaughlin further teaches a computer program comprises other web services (see at least section 1.2.2, particularly "web services"). However, McLaughlin does not explicitly teach a

script code section. Lucas teaches a *script code section* (see at least column 3, lines 45-55, particularly "XML-oriented language extensions for use in association with a scripting language"). Lucas further teaches *validating the script code ("JavaScript") section using the set of implementation instructions* ("XML data type declarations") (see at least column 3, line 56 through column 4 line 7, particularly: "a JavaScript-aware parser (e.g. parser 105) is equipped to recognize XML data type declarations and associate them with the appropriate items in the corresponding symbol table"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the constraint model of McLaughlin with the script code validation of Lucas because it would allow XML constraints to be used and tested with a script-based web service (see at least McLaughlin section 1.2.2 and Lucas column 3 lines 45-55).

Claim 2

Claim 2 includes all of the limitations of claim 1. Applicant states in page 3 lines 6-26 that their program is programmed using two separate tree structures, written in XML, wherein the first tree structure represents the classes to be implemented (definition module) and the second tree structure represents the associated interfaces (implementation module). McLaughlin further teaches *the set of definition instructions are definition modules* (see at least section 2.3.1, particularly "XML file" and figure 2-2 "XML documents"). McLaughlin further teaches *definition modules having a plurality of classes* (see at least section 3.3, particularly: "Once you've got your constraints ... you're ready to create a binding schema for your classes. This will instruct the class generation tool to generate classes, to use a specific Java package, to use collections, and a variety of other options." emphasis added).

McLaughlin further teaches *the set of implementation instructions are implementation modules* (see at least section 2.3, particularly "XML constraints" and section 6.4.4 "Interfaces, particularly: "To generate an interface, add this statement to your binding schema"). McLaughlin further teaches *implementation modules having a plurality of interfaces* (see at least section 3.2, "Creating the constraints", particularly "a set of constraints ready to generate classes from" and

example 3-1 “ELEMENT movies” and “ATTLIST movies version”). Because XML constraints are a model of the behavior of classes (“XML document”), they are an interface, as supported by specification page 3 lines 6-26.

Claim 3

Claim 3 includes all of the limitations of claim 1. McLaughlin further teaches *the set of definition instructions are converted into classes* (see at least section 3.1.4, figure 3-1 “Class generation process flow” and “The result of the generation step is one or more Java source files”).

McLaughlin further teaches *the set of implementation instructions are converted into interfaces* (see at least section 6.4.4 “Interfaces”, particularly “The result of this statement is a new generated class, the Person interface”).

Claims 4, 6, 10, and 11

Claim 4 includes all of the limitations of claim 3. Claims 6, 10 and 11 include the limitations of claim 1. McLaughlin further teaches *the set of definition instructions (“XML documents”) and the set of implementation instructions (“XML constraints”) are described in XML* (see at least section 2.3.1, particularly “XML file” and figure 2-2 “XML documents” and section 2.3, particularly “XML constraints”). It is readily apparent that McLaughlin teaches the files are in a tree structure because the files are XML files.

Claim 5

Claim 5 includes all of the limitations of claim 4. McLaughlin further teaches *the classes and the interfaces are defined in Java language* (see at least section 3.1.4, figure 3-1, particularly: “The result of the generation step is one or more Java source files”).

Claim 7

Claim 7 includes all of the limitations of claim 1. McLaughlin further teaches a computer program comprises other web services (see at least section 1.2.2, particularly “web services”). However, McLaughlin does not explicitly teach a script code section. Lucas teaches *the script code section is JavaScript* (see at least column 3, lines 45-55, particularly "a scripting language, such as JavaScript"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the constraint model of McLaughlin with the JavaScript script code validation of Lucas because it would allow XML constraints to be used and tested with a script-based web service (see at least McLaughlin section 1.2.2 and Lucas column 3 lines 45-55).

Claim 8

Claim 8 includes all of the limitations of claim 1. McLaughlin further teaches *validating the script code section comprises generating a symbol table by executing the code section in an interpreter (“parser”), and comparing the symbol table with the implementation instruction (“XML data type declarations”)* (see at least column 3, line 56 through column 4 line 7, particularly: "a JavaScript-aware parser (e.g. parser 105) is equipped to recognize XML data type declarations and associate them with the appropriate items in the corresponding symbol table"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the constraint model of McLaughlin with the script code validation of Lucas because it would allow validating a script-based web service with XML constraints (see at least McLaughlin section 1.2.2 and Lucas column 3 lines 45-55).

Claims 12-21

Claims 12-21 are a computer readable medium version, which otherwise recites the same limitations of the claims 1-8 and 10-11. The combination of McLaughlin and Lucas teaches all of the limitations of claims 1-8 and 10-11. It is readily apparent that the method taught by

McLaughlin includes instructions to implement the method on a computer readable medium (see, for example, section 3.1 step 4, "compile the classes").

Cited Prior Art

52. Hammerich et al. (US PG-PUB 2004/0123273 A1, hereafter '273) was cited in the Office Action mailed 8/28/2009 as claiming priority to the same European application (EP 02022042.2). Hammerich et al. (US 7,584,457) was cited in the Office Action mailed 8/28/2009 as being a granted patent from application '273.

53. **Examiner's Note:** The Examiner has pointed out particular references contained in the prior art of record within the body of this action for the convenience of the Applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply. Applicant, in preparing the response, should consider fully the entire reference as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner.

Conclusion

54. Any inquiry of a general nature or relating to the status of this application or concerning this communication or earlier communications from the Examiner should be directed to **Erika Kretzmer** whose telephone number is **(571) 270-5554**. The Examiner can normally be reached Monday through Thursday, 9:30am-6:00pm Eastern Time. If attempts to reach the examiner are unsuccessful, the Examiner's supervisor, **Tuan Dam** can be reached at **(571) 272-3695**.

Art Unit: 2192

55. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://portal.uspto.gov/external/portal/pair> . Please direct questions on access to the Private PAIR system to the Electronic Business Center (EBC) at **866.217.9197** (toll-free).

56. Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to 571-273-8300. Hand delivered responses should be brought to the United States Patent and Trademark Office Customer Service Window:

Randolph Building

401 Dulany Street

Alexandria, VA 22314.

/Erika Kretzmer/
Examiner, Art Unit 2192
February 23, 2010

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192